Session 127

Speech Recognition & Synthesis In-Depth

Kim Silverman, PhD
Principal Research Scientist
Manager, Spoken Language Technologies

Spoken Language Technologies Are in Mac OS X!

- Describe Speech Synthesis and Speech Recognition
- Guidelines about where and why to use them
- Lead you through the process of getting your applications talking and listening

Demo

Built-in Mac OS X Speech Technologies

What Speech Is in Mac OS X?

- Speech synthesis
  - Converts unrestricted texts to speech
  - 22 different voices
  - Currently U.S. English only
- Speech recognition
  - Speaker independent
  - Continuous speech
  - Far-field microphone
  - Currently U.S. English only

What's New in Mac OS X TTS?

- Merged Macintalk 3 and Macintalk Pro
  - Totally new codebase
  - Consistent behavior across voices
- All "Pro" voices are high quality
  - Crisper, more natural, more intelligible
- Improved pronunciation
  - Enlarged dictionary
  - Recursive morphological decomposition
  - Automatically trained letter-to-sound rules

What's New in Speech Recognition?

- Factored out pronunciation subsystem
  - Reduced RAM
  - More accurate recognition
  - Consistent behavior with TTS
- New UI
- Improved Speakable Items
  - Added XML-based command files

Why Use Speech?

- Increase the number of potential users
  - Younger users
  - Disability access
  - People less familiar with computation

Why Use Speech?

- Speech is a very natural form of communication
  - We've all been talking and listening all our lives
- Move beyond limits of "Point and Click"
  - You can't click on what you can't point to
  - "Click" = "grunt"
- Conversation is an appropriate modality for delegating tasks

What Are Some Ways I Can Use TTS in My Application?

- Notifications
  - Judiciously get the user's attention back
  - More informative than "beep"
  - Use for asynchronous events or events outside the user's control
- Additional feedback for younger users
- Proof reading

Accessibility

- TTS output for sight-impaired
- Augmentative communication
- Speech as an alternative control modality

Games!!!!!

- Speech control is cool
  - Enter cheat codes or global controls
  - Change views
  - Change weapons
  - Use point-and-speak: select forces and speak commands (e.g., “return to base”)
- Use for non-time-dependent control
- In noisy games, headsets might be needed

Education/literacy

- Interactive drills
- Teaching phonics
- Adult pronunciation improvement

Speech and Web Browsing

- Navigation within a browser
- Speak the links?
- Jump to new pages by topic
- Read out web pages
- Big Opportunity: VoiceXML

Form-filling

- An alternative to pop-up menus
- Use constrained language models for each field
  - "Create a new customer record"
    - "What type?"
  - "Corporate account"
    - "Payment schedule?"
  - "Thirty days"

Eyes-busy, Hands-busy

- e. g., Graphics
  - “Send to back”
  - “Increase brush size”

Demo

Sal Soghoian
AppleScript Product Manager

Delegation

- Use scripting or macros to combine multiple steps into one operation, with spoken dialogs to refine the task
  - Via AppleScript or within an app
  - Workflow automation
  - Home automation

Issues

- Educating users how to speak
  - No pauses, don't shout, . . .
  - Refer to "Helpful Tips" in Speech preference panel
- Background noise
  - Head-mounted microphones
- Non-native speakers
- Localization

It's Time to Code…
Speech Programming 101

Matthias Neeracher
Kevin Aitken

We're Built In!

- Speech Synthesis and Recognition are installed by default in Mac OS X
- Just link with Carbon.framework (or CarbonLib)
- Mac OS 9 APIs now accessible from Cocoa, Carbon, Classic
- Access via Obj-C, C, C++, Java, and more

Speech Synthesis

Simple Speech Synthesis

```
SpeakString("\pHello, World");
```

Flexible Speech Synthesis

- Open Speech channel

  NewSpeechChannel(&voice, &channel)

- Adjust parameters

  SetSpeechRate(channel, newRate)

- Speak

  SpeakText(channel, text, len)

There's More!

- Control over rate, pitch, modulation, volume
  - Customize the way speech sounds in your application
- Callback routines
  - Provide word and phoneme level feedback
- CocoaSpeechSynthesisExample illustrates many of these options

No Code Required

- Most speech controls can be *embedded* in text

```
SpeakString(
  “\pDon’t meet till [[emph +]] next Tuesday”);
```

Speech Synthesis at Its Best

Customize!!!

- Your app knows a lot more about how things should be spoken than the TTS engine does
- Filter the text (AAPL ⇨ “Apple Computer”)
- Customize the pronunciations
- Customize the intonation

Customizing the Pronunciation

- Use funny spelling?
  - **My name is Maw-tea-us**
  - Not recommended!
  - We may change way nonstandard spellings are pronounced
  - Unreliable and imprecise
- Use phoneme input
  - **My name is [[inpt PHON]] _m1AAt=tIY=UXs [[inpt TEXT]]**

Customizing the Intonation

- What is the problem?
  - Words alone aren't enough
    - "John only introduced Mary to Bill"
- Solution
  - Automatically determining the correct intonation is difficult in the general case
  - Application often can employ domain knowledge to do much better

Your first flight is with Alaskan Airlines flight 2762 departing from San Jose on Monday May 24th at 6:10 pm landing in San Francisco at 7:10 pm.
Thank you for choosing TTS Travel.

[[pbas +1; pmod +1]] **Your** [[emph +]] **first** [[emph -]] **flight: is with Alaskan Airlines: "**[[emph -]] **flight 2762".**

[[pbas -1; pmod -1; emph -]] **Departing from San Jose, on Monday May twenty fourth: at 6:10 pm,**

[[slnc 500; emph -]] **landing in San Francisco: at** [[emph +]] **seven** [[emph -]] **ten** [[emph -]] **p** [[emph -]] **m!**

[[slnc 600]] [[pbas +1; pmod +1]]

Thank you: for [[emph -]] **choosing TTS travel.**

1. Let the User Catch Up

- Add pauses at major "sense units" or boundaries of internal sentence structure
  - Liberally sprinkle punctuation wherever appropriate
  - Break up larger sentences in smaller ones
  - Lengthen more major pauses by adding extra silence
    - **[[slnc 500]]**

Departing from San Jose:
on Monday May 24th at 6:10 pm,
[[sInc 500]]
landing in San Francisco: at 7:10 pm.

2. Familiar Things Go in the Background

- De-emphasize repeated words

  …departing six ten…

  …landing seven [[emph -]] ten…

- De-emphasize items inferable from your overall application scenario

  Your first [[emph -]] flight is…

3. Liven It Up!

...landing in San Francisco at 7:10 pm!

4. Focus the User's Attention

- Use **[[emph +]]** on important or new information
- Pause with a colon just before the most important item

…at [[emph +]] seven [[emph -]] ten….

5. Use Paragraph Intonation

- Group sentences together into "Intonational paragraphs", based on topic structure
- Raise pitch range for first sentence, then reset

  [[pbas +1; pmod +1]] Your first flight…

  [[pbas -1; pmod -1]] Departing from…
- Put an additional pause between paragraphs

  …at 7:10 pm.

  [[slnc 600; pbas +1; pmod +1]] Thank you for…

Text to Speech Summary

- Customize pronunciation
  - Use phonemes for mispronounced words
- Customize intonation
  - Let user catch up
  - De-emphasize familiar things
  - Liven it up!
  - Focus the listener's attention on the important elements
  - Use Paragraph Intonation
- Give user a better speech experience

Speech Recognition for the Lazy

You've Got Choices

- Method #1: Add Speakable Items to your app
  - No speech code to write
  - Easy integration with Apple events
- Method #2: Use Speech Recognition API
  - Command lists can change according to context
  - Simple three step approach gets you listening fast

What Makes a Good Command?

- Like a menu item, but usually three to six words
  - Longer commands are more unique
  - Avoid single words like “hot”, “cut”, “quit” because they sound alike
- Testing
  - Make sure your commands are not easily confusable with other commands when spoken
  - Use Speakable Items or SRLanguageModeler to prototype your commands

Method #1: Add Speakable Items to Your Application

- Create and test your items
  - Say “Make this application speakable” to set up folder
- Add your items to your application bundle
  - Project Builder makes it easy to include the item files in your application bundle
- Lazily install items at run time
  - Supports self repair and self install
  - Great for multi-user environment

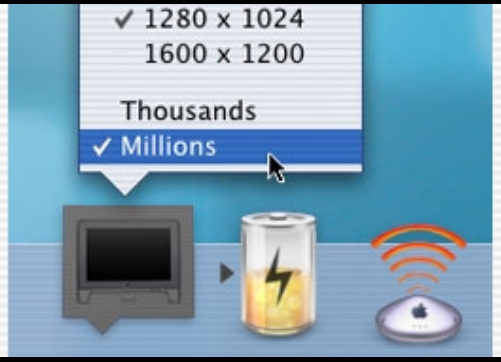
What's an Item?

- Any file that can be opened
- Two file types are most helpful:
  - Compiled AppleScript files
    - Use to send Apple events to your application
  - XML-based command files
    - Use to send keyboard events to activate menus and controls via keyboard shortcuts

Copy. Paste. Go!

- Use fauxTunes example as a starting point
  - Shows how to add spoken commands to a shipping product
  - Simplifies speech recognition programming for Cocoa and Carbon applications
- Download it from the web at:
  http://developer.apple.com/audio/speechrecognition.html

Example



Method #2: Use the Speech Recognition API

- Copy and paste from our examples to get started fast
- Coding approach is broken into three steps:
  - 1. Set up recognition objects
  - 2. Add commands and begin listening
  - 3. Handle recognition result

Simplified Recognition Process

Step 1:
Set up objects

Recognition System

Recognizer

Language Model

Step 2:
Add commands

Recognizer

Commands
- Speakable Items
- Applications
  - Switch to C P U Monitor
  - Switch to Chess
  - Switch to Finder
  - Switch to Grab
  - Switch to Help Viewer

Language Model

"Add to library"
"Make new playlist"
"Play this song"
"Turn visual on"

Step 3:
Handle result

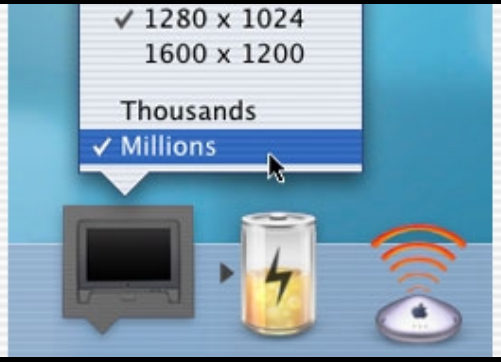
Speech Done
Apple Event Handler

"Play this song"

ID = 2

Play Song Routine

Example



Just Remember!

- Speakable Items support doesn't require additional speech code
- Listening for a list of commands using the Speech Recognition API is a simple three step process

Speech Recognition for Overachievers

Example: Chess

- Ships with Mac OS X
- Source code available at
  http://developer.apple.com/
  audio/speechrecognition.html
- Speech code is in ChessListener.cp
- Illustrates language model design

Language Is a Virus

- Unconstrained language model:

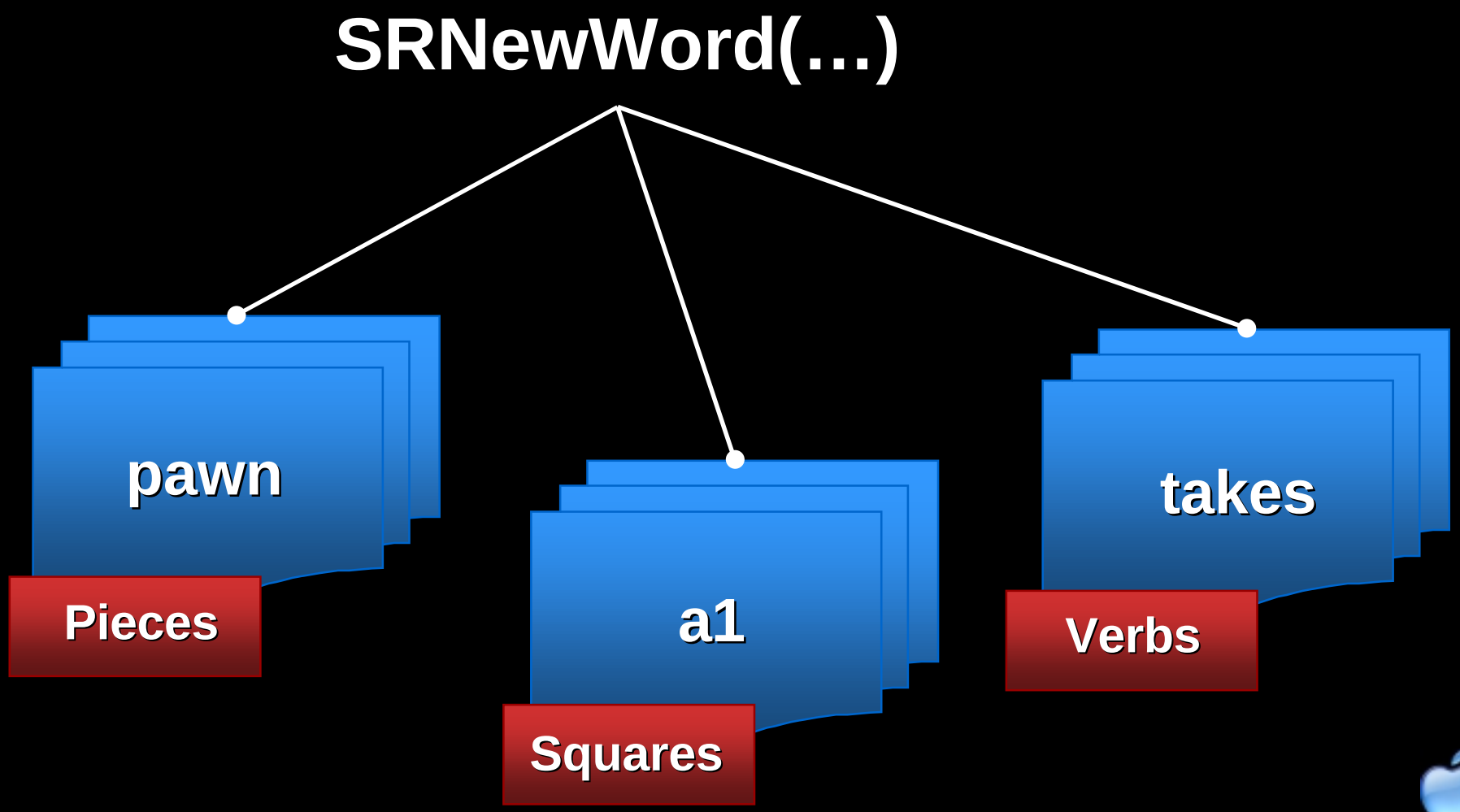
```
<move> = <piece> <square> to <square>
<piece>  = pawn | rook | knight
         | bishop | king | queen
<square>= A1 .. H8
```

- Total grammar paths: 6 * 64 * 64 = **24576** !

Constrain It!

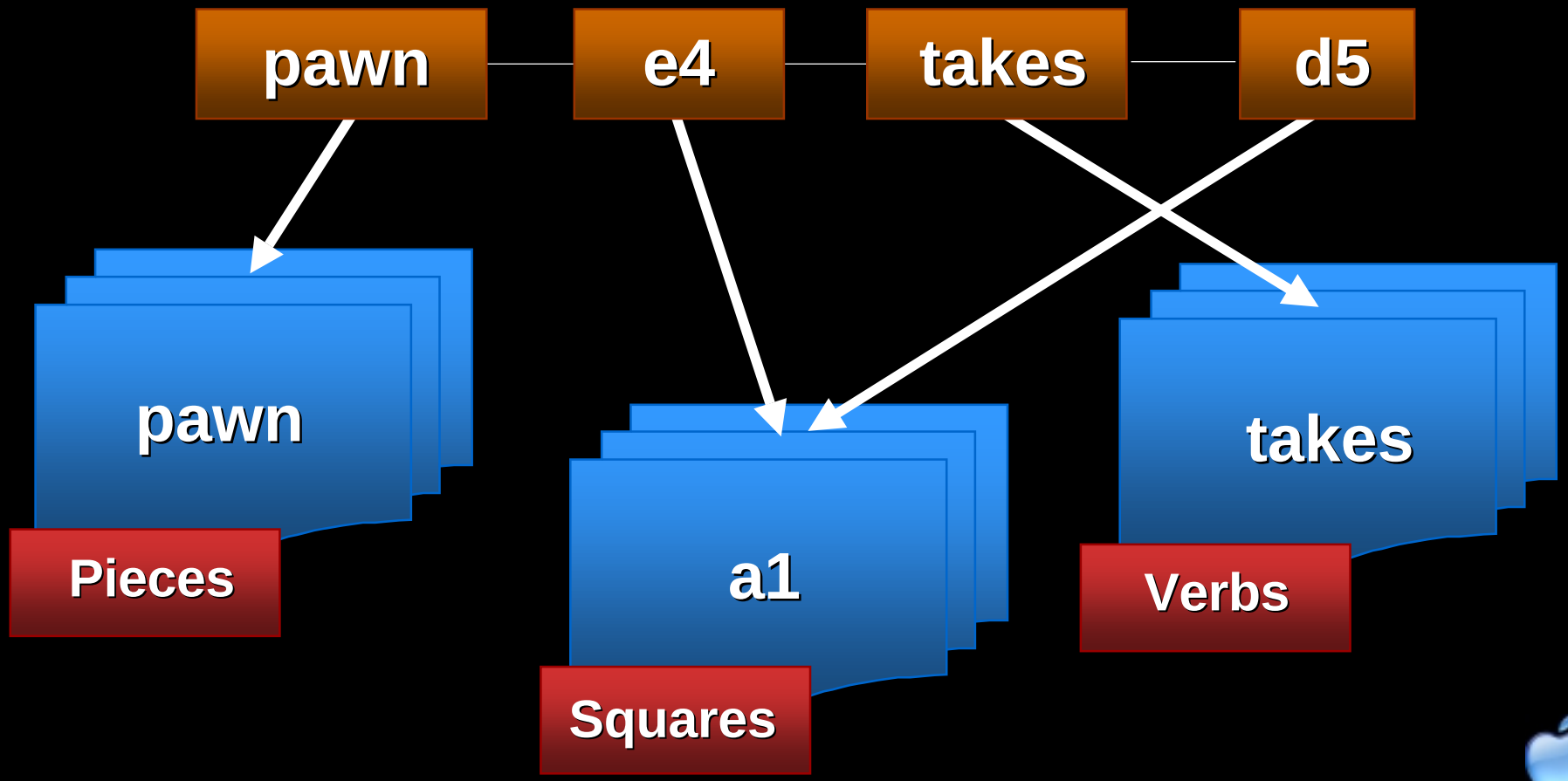
- In each chess position, only about 20 moves are actually legal
- So include only those!
- Don’t overconstrain:
  Leave in some plausible mistakes

Prefabricated Parts



Some Assembly Required

SRAddLanguageObject(...)



Make Your Move

- Constrain your language model to plausible commands
- Update it when conditions change
- Consider using prefabricated language objects

Building Language Models

- SRLanguageModeler tool lets you experiment
- Live tests for rapid turnaround
- AIFF batch tests for systematic testing

What Sample Code Is Included?

- TTS Samples
- ASR Samples

/Developer/
Examples/
Speech/

Now We Are Talking!

- Speech Synthesis and Speech Recognition are part of Mac OS X
  - Conceptual overview of APIs
  - Why you want to use them
- Gave guidelines on how to use them well

Speech-enable Your Apps

- Good
  - SR: Speakable visible controls
  - TTS: Speak simple alerts and panels
- Better
  - Delegation: Group a number of steps into a single spoken command
  - TTS: Customize texts, read back information
- Best
  - Interactive spoken dialogs
  - Form filling

Roadmap

119 Carbon Controls & Appearance I	Hall 2 **Thurs., 9:00am**
120 Carbon Controls & Appearance II	Hall 2 **Thurs., 10:30am**
130 Designing & Using Aqua Icons	Hall 2 **Fri., 10:30am**
125 Apple Help	Room J2 **Thurs., 2:00pm**
114 Application Packaging & Document Binding	Hall 2 **Wed., 5:00pm**

Roadmap

702 Interface Builder	Hall 2 **Tues., 3:30pm**
FF103 Feedback Forum: Aqua	Room J1 **Wed., 3:30pm**
FF114 Feedback Forum: High Level Toolbox	Room C **Fri., 3:30pm**

Who to Contact

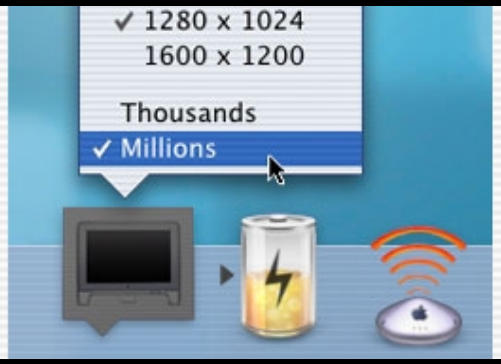
Technical questions?
Developer Technical Support
dts@apple.com

For more information:
Inside Mac:
Speech Manager
Speech Recognition Manager
WWW: **http://www.apple.com/speech**

To register as a volunteer for data collection:
plaintalk@apple.com

http://developer.apple.com/events/urls.html

Q&A



Heather Hickman
Cocoa Technology Manager
hhickman@apple.com

http://developer.apple.com/events/urls.html

WWDC
Worldwide Developers Conference 2001